

Amendments to the Claims

(None)

1. (Previously presented) A program execution method for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, comprising the steps of:

optimizing the loop process, said optimizing step including the steps of:

moving said one or more transfer points to the top of said loop process if they can be moved there without a problem occurring;

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process;

storing information for generating recalculation code for one or more specific transfer points when privatization, common sub-expression elimination, and moving of code that are performed pass beyond said specific transfer points; and

performing a recalculation during a transfer process; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

2. (Previously presented) The program execution method according to claim 1, further comprising the step of:

defining as a new transfer point, a point from said interpreter process to said compiled code process whereat, when said method that is currently being executed is replaced, the execution speed is increased compared with when said method is not replaced.

3. (Previously presented) The program execution method according to claim 1 or 2, further comprising the steps of:

generating information required to perform a transfer from said interpreter process to said compiled code process; and

storing said generated information while correlating said generated information with said transfer points,

wherein, at said recalculation step, said information stored for said transfer points is employed.

4. (Previously presented) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, the method steps comprising:

optimizing the loop process, said optimizing step including the steps of:

moving said one or more transfer points to the top of said loop process if they can be moved there without a problem occurring;

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process;

storing information for generating recalculation code for one or more specific transfer points when privatization, common sub-expression elimination, and moving of code that are performed pass beyond said specific transfer points; and

performing a recalculation during a transfer process; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

5. (Previously presented) A program execution method for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, comprising the steps of:

optimizing the loop process, said optimizing step including the step of moving said one or more transfer points to the top of said loop process if they can be moved there without a problem occurring; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

6. (Previously presented) The program execution method according to claim 5, said optimizing step further including the step of:

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process.

7. (Previously presented) A program execution method for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, comprising the steps of:

optimizing the loop process, said optimizing step including the step of copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

8. (Previously presented) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, the method steps comprising:

optimizing the loop process, said optimizing step including the step of moving said one or more transfer points to the top of said loop process if they can be moved there without a problem occurring; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

9. (Previously presented) The program storage device of claim 8, said optimizing step further including the step of:

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process.

10. (Previously presented) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for transferring, from an interpreter process to a loop process of a compiled code process, a method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, the method steps comprising:

optimizing the loop process, said optimizing step including the step of copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.